AMASON: Abstract Meta-model for Agent-Based SimulatiON

Franziska Klügl[1] and Paul Davidsson[2]

[1] School of Science and Technology
Örebro University, Örebro, Sweden
`franziska.klugl@oru.se`
[2] School of Technology
Malmö University, Malmö, Sweden
`paul.davidsson@mah.se`

Abstract. The basic prerequisite for methodological advance in Multi-Agent Based Modelling and Simulation is a clear, ideally formally-grounded, concept of our subject. A commonly accepted, implementation-independent meta-model may improve the status of MABS as a scientific field providing a solid foundation that can be used for describing, comparing, analysing, and understanding MABS models. In this contribution, we present an attempt formalizing a general view of MABS models by defining the AMASON meta-model that captures the basic structure and dynamics of a MABS model.

1 Introduction

In this contribution, we present AMASON (Abstract Meta-model for Agent-based SimulatiON) as a general formalization of Multi-Agent Based Simulation (MABS) models. As the example of Agent-Oriented Software Engineering (AOSE) shows, a well-defined meta-model provides a solid foundation for all processes relevant for "engineering" MABS models: specification and documentation, appropriate tool support, teaching and exchange and reuse of models. A meta-model provides a clear set of categories and a language for describing a MABS model. It can be used in combination with documentation frameworks such as ODD [14]. ODD gives a structure for documentation, yet no clear language. Describing different MABS models using concepts taken from the same meta-model supports their comparison and comparative analysis, especially if the models are described without programming language level details. This argument that can be also found in [20] suggesting the use of explicit model ontologies. A prerequisite is hereby that the meta-model is formulated in an implementation- and platform-independent way.

Last but not least, teaching MABS development is facilitated if a concise meta-model is available, as fuzziness of terms and structures are avoided. MABS could be handled and introduced in the same rigorous way as other microscopic simulation paradigms with a clear definition of their basic structures. Practical introductions such as [27] could be augmented with a theoretical introduction stating which elements a MABS model has and how they relate to each other. Clearly describing the scope of MABS

M. Klusch, M. Thimm, and M. Paprzycki (Eds.): MATES 2013, LNAI 8076, pp. 101–114, 2013.

and also clearly stating what MABS models are not, is supported by an appropriate meta-model.

Thus, the existence of a shared meta-model would improve the status of MABS as a scientific and engineering field. Within the scope of this contribution with its limited number of pages, we can just sketch the first considerations towards such a meta-model, called AMASON. Many aspects that are especially important for the social sciences, such as negotiations, norms and organization, are left to future work, as we first wanted to find an abstract and general common ground that also includes domains beyond social science simulation. Moreover, we do not want to define the meta-model on a too detailed level including e.g. data types, but instead throw light on the really important aspects. Nevertheless, a generic, abstract approach leaves space for more specific extensions and specializations.

In the remainder of the paper we will first give the background for our proposal, discussing mainly related work from AOSE and MABS. In section 3, we introduce the basic structural and dynamic elements of the meta-model in its current status. We end by wrapping up and discussing future work. Due to space limitation, we cannot give a consistent full example using the meta-model. Yet, the description of the meta-model is illustrated with the basic Sugarscape model [6] as running example.

2 Background and Related Work

2.1 Meta-models in Multi-agent Software

Introductory literature on multi-agent systems such as [32] uses formal descriptions of what is an agent, how it is embedded in an environment and indicate abstract architectures. Although apt as a basic introduction to agents (in form of a one-agent system), the value of these descriptions for capturing the idea of multi-agent systems is limited. A more elaborated formulation of meta-models for multi-agent systems can be found in [5] and its successors using *Z* schemata starting from the definition of an "object" refining it to "Agent" and "Autonomous Agent". Their meta-model is very detailed also including statements on goals and motivations, commitments and obligations and relations between agents in a multi-agent system with respect to collaboration.

Concisely defined meta-models for multi-agent systems play an important role in the area of AOSE. A meta-model hereby defines the framework of concepts and their relations providing a language for analysis and specification. Their clear and formal definition received a lot of attention especially together with topics such as model-driven design or method fragmentation. Only, if the meta-model is absolutely clear, interfaces between models on different abstraction levels or for different aspects can be created or combined. Thus, it is not surprising that there is a wealth of meta-models used in AOSE. UML-based specifications of meta-models for ADELFE, Gaia and PASSI can be found in [2]. The AOSE methodology INGENIAS and its meta-model have been successfully applied to agent-based social simulation [9].

Hahn et al. [15] introduce a platform-independent meta-model integrating different views from an multi-agent view to an environmental view. Also, meta-models have been defined specially for the specification of organizational concepts; see [7] for the ALAADIN meta-model providing the terms agent, group and role as core organizational concepts. Also Beydoun et al. [3] aim at providing a general meta-model and hence a modelling language that can be used for describing agents independent from an application.

Yet in all those AOSE meta-models focussing on organizational concepts, (virtual) time and spatial environment play none or just a subordinate role, but would be central in simulation settings. A model that clearly shows the relevance of the part of the overall system representing the environment is the “influences and reactions” model [8]. It gives a precise understanding of the relation between the agents’ actions with the agent want to influence its environment and the reaction of the environment that finally determines what the agents’ action is actually affecting. Recently, it has been further developped into a general concept of interaction in the MASQ meta-model [4]. In addition to the Influence-Reaction principle, R. Dinu et al. hereby postulate the separation of mind and body, as well as a segmentation of space (and four more principles including culture). Currently, MASQ is more a family of meta-models avoiding particular design choices.

Another meta-model with a strong focus on environmental entities is the Agents& Artifacts model [25] with a clear distinction between agents and “reactive objects” that – as explicit environmental entities – provide services and thus support the interaction between the actual agents. Thus, the A&A meta-model forms an abstraction from the real environment towards a framework that supports the design and implementation of Multi-Agent Systems.

There are several reasons why the AOSE meta-models are not appropriate for describing MABS models. A fundamental reason is the fact that a MABS model is a virtual representation of another system (including individuals, objects, etc.), whereas a MAS is an artifact that interacts with an environment. The meta-models on which the different AOSE methodologies are based, nevertheless can be valuable as they provide concepts for analysing, describing and specifying multi-agent models. Yet, a modeller must be aware about the assumptions that the usage of those meta-models created for supporting the design of a Multi-Agent System impose on the final model. Meta-models which are based on grounded social science theories of institutions or similar would take that burden from the modeller.

2.2 Conceptualizations in MABS

There is clearly a lack of formal specification in MABS. Introductory textbooks such as [13], [23] or [27] are based on heuristics and best practices. They do not give a formal and precise grounding beyond a textual characterization what a MABS model is and may contain. Similarly, the ODD Protocol [14] provides a framework for documentation that clearly advices which elements are necessary for a full documentation. However, ODD is not giving a meta-model or language that a modeller can use for capturing the elements of the MABS in the different submodels. Bandini et al. [1] give a good conceptualization of elements of multi-agent systems relevant for MABS. Yet, without formalization their approach remains nevertheless fuzzy in detail.

During the last years, a number of approaches have been published with a similar objective as in this contribution. An early meta-model can be found in [18]. There, an explicit separation between resources and agents is made which is distinct from our more transparent characterization of bodies and minds. Another important difference is the environmental structure that has not been elaborated earlier.

The reference model proposed in [28] is fully based on an event-based approach, in which sensor and effector activation create cascades of events that may manipulate the internal state of the agent as well as its external environment. Constraints are used for representing events that do not work in a particular environmental configuration. One aspect that [28] explicitly points out is that actions – which are conceived as activation of effectors – take time. The only atomic element in the reference model is the event.

There are several formal frameworks for representing simulation models starting from the system science-inspired DEVS framework of Object-Oriented Simulation [33] Approaches such as AgedDEVS [29], form the formal model underlying the JAMES II system. DEVS/RAP [17] or PECS [30] reference model can be seen as extensions of Object-Oriented Simulation models integrating more or less sophisticated agent structures for capturing internal agent processes, belief or plan structures as well as variable overall system structures with agents that are generated, die or change interaction partners. Mueller [22] systematically analyses which extensions of the DEVS meta-model for event-based Object-Oriented Simulation [33] capturing parallel simulation and variable structures, can best serve as a basis for MABS models. An advantage of DEVS is the precisely defined semantics of basic elements and different update functions. Due to its generic nature based on the notion of "system" with input, internal state and output, it can be seen as forming the starting point for many, if not all meta-models in agent-based simulation.

A generic formalization of MABS concepts has been suggested by Helleboogh et al. [16]. They focus on interactions between agents' activities and an explicit environment. Dynamism is elaborated based on the concept of activities. Activities represent the agents' influences, "reaction laws" determine how the environment reacts on those and transforms them. Interaction is handled using "interference laws". The overall goal is to clearly define environmental dynamics. Also [31] use the influences and reactions idea of [8] as a starting point, but focus on non-global synchronization as it is necessary for truly distributed applications.

Recently, two suggestions for MABS meta-models have been published that can be seen as attempts to develop meta-models similar to the AOSE meta-models, yet adapted to challenges of (social science) MABS simulation: MAIA [12] and easyABM [10] provide languages with a focus on the societal level. MAIA formalizes social institutional theories, easyABM provide an overall detailed view that is apt for code generation yet shares many problems with the traditional AOSE approaches sketched above.

Thus, there is currently no meta-model that provides a basic view on what a MABS model is and contains, expressed using a minimal set of concepts that is applicable to all types of models – ranging from social science to models with simpler agents.

3 A Generic Meta-model: AMASON

AMASON, the meta-model that we propose originates from our collected experience of building and analysing MABS models in various domains. As we aim at a simple model that makes the core concepts explicit, we cannot take a meta-model of for example object-oriented simulation and add concepts. Also, existing meta-models for MABS platforms, such as [26] or [24] are far too detailed and specific containing low level data types.

AMASON contains two basic views: the context of the model and the contents of the model. In the following we focus on the model itself. The context of the model is then needed for setting up a simulation based on the model. It is essential for developing, validating and using the model. The model context contains information about the objectives, how the model shall be used and under with which parameter configurations it can be used, information on how and with what data the model is validated, etc. Whereas these aspects are relatively obvious considering standard simulation literature, conceptual confusion occurs when one attempts to come up with a underlying conceptualization of the actual MABS model. Despite of their importance for documenting the model, in this contribution we concentrate on the core content of the meta-model as this can be used for formulating a more structured context model.

3.1 Elements of a Model

We identified three types of components for a multi-agent simulation model: $Body$, $Mind$ and $Regions$. All three may possess some form of internal state. The idea behind is to separate the physical entity from the mental one to make embodiment explicit and also provide a clear distinction between entities that possess reasoning capabilities and entities that do not, but nevertheless populate the environment and are of use (or obstacles) for the active entities capable of decision making. The artefact framework [25] shows that this conceptual separation is also relevant for multi-agent system software. The idea behind regions is to provide a uniform perspective on the large variety of spatial environments. Whereas a body is located in a region, a mind needs to be connected to a body for achieving situatedness. An agent consists of both body and mind.

Body. A body represents a physical entity in a MABS model: A human or an animal body, the physical parts of a robot, but also rocks, food items or houses. The body forms the "hardware" of an agent. It carries sensors and actuators and thus provides the technical means to interact with the environment. Thus, it is the body which is located on a specific region.

Each body has a state that is domain- or model-specific. The state may be structured into a set of parameters and state variables that may contain arbitrary complex data ranging from numbers denoting energy storage to structure representing a complex metabolism. States do not need to be discrete, yet the body is in a particular state at a particular time in a particular simulation run.

The state of a bodycan be updated by region-specific processes, which are captured by the state of the region. For example the temperature of a bodymay rise if the temperature of the regionis high. Such state changes may happen without any decision making of a mind. In a more formal treatment: the set of all body-elements in a simulation model is $B = \{b_1, b_2, ..., b_n\}$ with b_i as an individual body with index i. Every body has an individual set of possible internal states described by Σ_{b_i}. The state of body b_i at time t is denoted with $s_{b_i}(t) \in \Sigma_{b_i}$. The initial state is $s_{b_i}(t_0) \in \Sigma_{b_i}$ where t_0 is the point in time when b_i is generated.

Mind. To become an (intelligent) agent, a body needs to be coupled to a mind. This entity contains reasoning capabilities, thus handles the decision-making processes. It possesses also an internal memory or state whose structure and content is depending on the particular reasoning mechanism. As in the body, we do not restrict the structure or representation languages used for expressing a mind state. A typical structure might be a BDI architecture, the state would then contain current beliefs, goals, and committed plans. In a different model instantiation, a mind may be based on a neural network - the weights of the neural network then would correspond to the current state of the mind.

We denote the set of all minds with the letter $M = \{m_1, m_2, ..., m_k\}$ with m_j as an individual mind with index j. The number of minds $k \leq n$ with n as the number of bodys. Every mind may have an individual set of possible internal states Σ_{m_j} being the set of possible states of mind with index j. The state of mind m_j at time t is denoted with $s_{m_j}(t) \in \Sigma_{m_j}$. The state hereby is depending on the particular reasoning approach that the mind m_j uses (see section 3.2). The initial state is $sm_{m_j}(t_0) \in \Sigma_{m_j}$ where m_j denotes the particular mind, t_0 is the point in time when m_j is generated.

We call the coupling between body and mind the "embody" relation: $embody : M \rightarrow B$ with M as the set of minds and B as the set of bodys. A body without a mind corresponds to a passive object in the environmemnt.

Region. A central part of a MABS model is the spatial environment where the agents and objects are situated. A large variety of spatial representations is in use: models with discrete of continuous maps, cellular automata, network structures with or without metrics; there are also models without explicit space. In all cases (except those in which all agents are virtually at one location, which in [21] are called "aspatial soups"), an explicit environmental structure is necessary for supporting the representation of locality. We suggest the idea of connected regions as a general, yet structured way of conceiving a heterogeneous model of the spatial environment of a MABS model.

Every region has its specific spatial representation (continuous, discrete or aspatial). A region is conceived as an explicit entity with a state. Thus, possibly heterogeneous global properties, such as temperature or light can be captured in a way similar to a body.

Connections between the regions form the edges of a network of regions. They may be dynamic - that means they may be created by agents or destroyed. How a connection looks like, is depending on the particular spatial model used in the regions. For instance, a connection may be a (directed) link without structure or it may be a door with a given width. It is quite obvious how continuous or discrete maps can be conceived as one region or that a pure network corresponds to connected regions, where one region is one aspatial node of the network. Yet, the concept also supports more complex spatial structures:

- Grid maps consist of cells that carry information which is accessible for the agents located on a cell. This information may be a discretized gradient field for a pedestrian simulation. Each cell would hereby correspond to a region, its state represents the gradient value. Every cell-region has connection to its neighbouring cells. The cell itself has no map, all agents on that cell have no further detailed position. A cellular automaton can be conceptualized by adding an agent to each cell which does not move, but intentionally updates the state of the cell.
- A second interesting case are metric networks: network structures in which the nodes have positions in some metric space and the length of connections between nodes are relevant for the agents populating such an environment. Examples are road networks. For expressing such an spatial structure, a network of connected regions needs to be located within on higher-level region containing a continuous or discrete map.

The consequence of these considerations is that the meta-model should allow for hierarchies of regions so that, e.g., a network of regions can be located on/in a region. This regions within regions concept together with the explicit connection model forms the appropriate meta-model for capturing all possible environmental representations that we encountered so far.

Capturing regions in a more formal language gives: As for bodys and minds, the set of all regions is $R = \{r_1, r_2, ..., r_l\}$ with r_x as an individual region with index x. The state of region r_x at time t is denoted with $s_{r_x}(t) \in \Sigma_{r_x}$ – similar to the body.

Regions are connected with explicit connections. The structure of a connection is depending on the particular structure of the region: if the region is a container without internal spatial structure, the connection corresponds to a link; if the region has a 2-dimensional grid structure, a connection maps a set of grid cells on region r_x to a set of grid cells on region r_y. If the region has a 2-dimensional continuous map, a connection maps a set of positions (line) at the edge of one region to the other. In case of a 3-dimensional map, the connection may not just be a line, but could be also a part of plane. Also connections between different forms of regions are possible, such as connecting a 2-dimensional grid (e.g. representing some ground surface) to a 3-dimensional continuous map (e.g. representing the atmosphere). Thus, we define a "connection area" C which's particular form depends on the regions that it connects. Connection are then represented by a function $connect : R \times R \rightarrow C$.

The $locate$ function assigns a position in a region to a body: $locate_{r_x} : B \rightarrow P_{r_x}$. The $locate$ function is depending on the particular region. P_{r_x} is the set of possible positions in that region r_x. What a position can be is depending on the region. For a continuous region with m dimensions, the set of possible positions would be $P_{r_x} = \mathbb{R}^m$.

Figure 1 gives an overall summary of the different elements of the AMASON meta-model.

Illustration: Sugarscape. We use the famous Sugarscape model for illustrating the elements of AMASON presented so far. In the Sugarscape world (we refer to the initial,

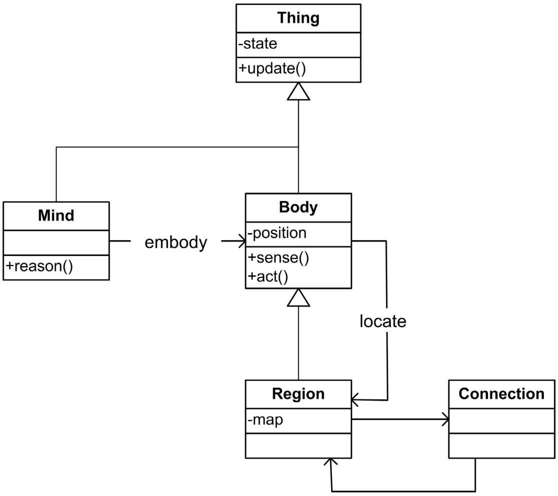

Fig. 1. Overview over the elements in AMASON

simple version that is described in chapter 2 of [6]), agents move on a grid harvesting and consuming "sugar".

Following the concepts above, a region corresponds to a sugar cell. All region-cells possess of same structure, yet not the same state: The state of a region is expressed by a vector of one variable and two parameters: $CurrentSugar$, $GrowthRate$ and $SugarCapacity$. As these are modelled as natural numbers, the set of possible states is: $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$. The position of each cell is determined implicitly by its connections to its four neighbours (as we assume a torus, all regions have four neighbours). Explicitly modelled, there are four connections: c_{west}, c_{east}, c_{south} and c_{north} for the respective neighbouring relation. An explicit grid position would help setting up the landscape. The initial state values for the variables are given by the two sugar hills.

Agents – consisting of a body and a mind – are located on a region. More precisely: The body of an agent is located on a region. There is just space for one body on a cell. Thus, the $locate : B \rightarrow R$ is assigning one bodyto one region. We assume that all agent parameters and variables are assigned to the body, representing the metabolism of sugar consumption. The mind is responsible for the decision making about where to go next. The state of a Sugarscape Body is captured by two constants and one variables: $PerceptionDistance \times Metabolism \times SugarStorage$. The constants are parameters with individual values for each agent, but an agent cannot change them. Assuming that perception is restricted to a max limit of 6 cells and that neither the consumption of sugar nor the potential storage of sugar is limited, the set of all possible states of a Sugarscape body Σ_{b_i} is: $\{1, 2, 3, 4, 5, 6\} \times \mathbb{N} \times \mathbb{N}$. The position of the body can be determined via the $locate$-function. Hereby, we leave open whether in praxis, the body stores the connection to the region or the region possesses a list of bodys on it. Due to the simplicity of this agent, its mind has an empty state. Later we will introduce a variable with temporary values of cells for facilitating the agent dynamics. Real extension involving e.g. Trade would require parameters (thresholds when to sell) and variable information on the budget.

3.2 Dynamics

Based on the basic structural concepts presented above, the relevant processes are defined: For expressing dynamics, an explicit representation of time is necessary: T is the time domain. The nature of the time set influences how the model can be executed. A usual choice is $T = \mathbb{N}$ with state changes only at discrete points in time. Whether a model is time- or event-driven is regarded as an implementation issue. The AMASON meta-model should be independent from how a simulation is implemented.

Dynamics in a MABS model can be associated with the following elements: a) the environment dynamics that cannot be influenced by the agents (cf. exogenous processes and events), b) the agents that intentionally manipulate their environment and themselves.

Environmental Dynamics. Environmental dynamics happen without being triggered by an agent. Processes such as seasonal temperature dynamics, a tree growing, or rain starts to fall or a stone is heating up are examples. In the meta-model we associate such dynamics with region s. One can distinguish between dynamics that just affect the state of the region, and dynamics that affect the state of bodys that are located on the region:

- Let $s_{r_x}(t) \in \Sigma_{r_x}$ be the state of the region r_x at time t. Then, we can capture the purely region-related dynamics by: $updateRegion : \Sigma_{r_x} \times T \rightarrow \Sigma_{r_x}$. With this function the state of the region changes over time without external triggers. With this, seasonal dynamics can be expressed. With humidity as a variable of the region, also the effects of events such as rainfall can be captured.
- Let Σ_{r_x} be the set representing possible states of the region r_x and $B|_{r_x}$ are the bodys that are currently located on r_x. The function $updateBodyState : \Sigma_{r_x} \times \Sigma_{B|_{r_x}} \times T \rightarrow \Sigma_{r_x} \times \Sigma_{B|_{r_x}}$ updates the state of all bodies that are located on the region based on their previous state and the state of the region. This is the right level to formulate that the body of a stone heats depending on the temperature of the region, on which it is located. With mobile agents the domain of the function is changing depending on the bodys that are at each time on the region, so a precise, closed mathematical formulation may be difficult.

Agent-Based Dynamics. Following the general concept of an agent, there is no doubt that agent dynamics follow a *sense-reason-act* process combination. This must be integrated with the distinction in body and mind. As they belong to the physical part of an agent, sensors and effectors are associated with its body yet reasoning with the mind. As "interfaces" between sensing and reasoning, we conceive "perception" that translates the sensed data into information that is relevant for reasoning. The reasoning then produces actions. Depending on the body state actions are transformed into executable actions (commands of the effectors) which then are handed over to the region, on which the body is located. The region is responsible for producing the actual effects of all actions of agents on it. More formally, agent-based dynamics involves processes at bodys, minds and regions:

- Body: $sense_{b_i} : \Sigma_{R|_{b_i}} \times \Sigma_{b_i} \times T \rightarrow \Sigma_{b_i}$ with $R|_{b_i}$ denotes a set of regions that are currently accessible for the sensors of b_i. This is a simplification replacing an elaborated concept of a local environment. Movement of the agent again would result in a varying definition of the $sense$ function, as different regions might be relevant. The region on which the body b_i is located should be part of this set. The results of the sensing process may be stored in a sensor memory with particular states as well. For reasons of simplicity, we assume that this sensor memory is part of the physical state of the body.
  $act_{b_i} : Act_{m_j} \times \Sigma_{b_i} \times T \rightarrow ExecAct_{b_i}$ is the function that transforms actions Act_{m_j} as instructions from the mind m_j that is controlling the body b_i ($embody(m_j) = b_j$) to actions $ExecAct_{b_i}$ that the body can actually do depending on its current state.
- Mind: The mind m_j possesses three processes that mirror the classical *perceive-reason-act*: $perceive_{m_j} : \Sigma_{b_i} \times \Sigma_{m_j} \times T \rightarrow \Sigma_{m_j}$ with $embody(m_j) = b_j$, processes the sensor data stored in the state of the body to information relevant for the mind. With this information, the current state of the mind is also updated in $update_{m_j} : \Sigma_{m_j} \times T \rightarrow \Sigma_{m_j}$ expressing some internal reasoning. Then the mind selects an action: $action_{m_j} : \Sigma_{m_j} \times T \rightarrow Act_{m_j}$. With Act_{m_j} as the set of all possible actions that the mind m_j may come up.
- Region: When the effectors of the body b_j execute a given executable action, this does not automatically mean that the environment changes. The actual effect on the environment is handled by the region r_x, which takes all the executable actions of all bodys that are located on it, for determining the actual effect of the combined actions: $effect_{r_x} : ExecAct_{B|_{r_x}} \times \Sigma_{r_x} \times T \rightarrow \Sigma_{r_x}$.

Illustration: Sugarscape Continued. Starting from the above given structure, the next step is to specify the different functions describing the dynamics: We start with the environment: The increase of the $CurrentSugar$-Value by the growth rate upto the capacity of the cell. $GrowthRate$ and $SugarCapacity$ are constants.

$$\begin{aligned} &updateRegion(< currentSugar, growthRate, sugarCapacity >, t) = \\ &\qquad < min(currentSugar + growthRate, sugarCapacity), ., . > \end{aligned}$$

This is followed by the regions-specific influence on bodys. This contains two elements: the consumption of sugar in the body metabolism and the harvesting of sugar. The $SugarStorage$ of the body is reduced by the value of $metabolism$, but increased by the $currentSugar$ of the cell on which the body is located. The value of the $currentSugar$ variable of the cell is set to 0:

$$\begin{aligned} &updateBodyState(< currentSugar, ., . >, < ., metabolism, currentSugar >, t) = \\ &(< 0, ., . >, < ., metabolism, (currentSugar - metabolism) + r.currentSugar >) \end{aligned}$$

If the $sugarStorage$ of the body becomes negative, the region deletes the body together with its corresponding mind.

For deciding where to go next, the agent must first determine the subset of sugarcells that it can sense. The size of that set of cells forming the domain of the $sense$-function, depends on the value of $PerceptionDistance$. So with $PerceptionDistance = 2$, the domain of the $sense_b$-function is a set of cells with $c_b = locate(b)$: $\{b, c_{west}(b), c_{west}(c_{west}(b)), c_{south}(b), ...\}$. For storing the result of sensing, we introduce the before mentioned variable b containing a datastructure with a table of the current sugar values in the neighbourhood with the direction towards them: $SugarPercept$.

The $percept$ function of the mind transfers the information in the $SugarPercept$ variable to a corresponding one in the mind for making the agent aware of that information. For reasons of clarity, we introduce an additional variable supporting reasoning that stores the direction towards the highest amount of $CurrentSugar$ out of the perceived ones. This variable is set in the $update_{mind}$ function. The domain of the state variable $Direction$ is $\{$ *west, south, east, north, 2west, 2south, 2east, 2north*$\}$ (for the agent which's body has a perception radius of 2.

The action-selection function selects the action of moving into the optimal direction: $Act_m = \{$*Noop, West, South, East, North, 2West, 2South, 2East, 2North* $\}$ encoding direction and speed. In this simple version of the Sugarscape model, the action that the mind selects is also the one that the effectors of the body send to execution to the cell on which the body is located. The cell on which the body is located, will then move the body (together with its mind) to the envisioned cell, if the cell is not occupied.

3.3 Putting Elements together

Combining structural information and dynamics, the dynamics around minds and bodys must be more elaborated: A model of a body contains thus not only the state and the initial state, the set of actions that it can execute, but also the particular $sense$ and act functions:

$$b_i = \langle \sigma_{b_i}, s_{b_i}(t_0), ExecAct_{b_i}, sense_{b_i}, act_{b_i} \rangle$$

Consequently, a similar collection can be done for the minds:

$$m_j = \langle \sigma_{m_j}, s_{m_j}(t_0), Act_{m_j}, perceive_{m_j}, update_{m_i}, action_{m_i} \rangle$$

The full description of a region contains the information of the state of the region a characterisation of its spatial representation, as well as a map-specific function for localisation. Then, there are three functions describing dynamical processes under control of the region: the independent dynamics of the region, the update of the body states depending on the region and the effect function executing the agents' actions.

$$r_x = \langle b_{r_x}, map_{r_x}, locate_{r_x}, updateRegion_{r_x}, updateBodyState_{r_x}, effect_{r_x} \rangle$$

So, a MABS model consists of a population of bodys without and with a mind and a representation of the spatial structures determining the basic structure of the shared environment. Thus, the population of entities consists of all bodys, all minds and the $embody$ relation connecting them: $POP = \langle B, M, embody \rangle$ with $B = \{b_1, ..., b_n\}$

the set of all bodys, and $M = \{m_1, ..., m_k\}$ the set of all minds. Similarly, we collect all information on the environmental structure in $ENV = \langle R, C, connect\rangle$, combining all regions and their connections, that means the elements that describe how a number of regions are connected and the function that actually maps them together. The overall model is then defined combining population, environment and the $locate$-function connecting the bodys and the regions: $MABM = \langle POP, ENV, locate\rangle$. The overall $locate$-function has the region- specific $locate$-functions as elements.

This overall division into population and environmental structure reduces the environmental model of the MABS model to an active spatial structure. This is at first sight a little bit counter-intuitive and contradicts e.g. [19] who argue that the simulated environment may contain also explicit static entities. In our conceptualization the static elements are conceived as bodys without minds and are thus part of the population. We intentionally decided for this as a distinction between core agents and environmental agents is somehow artificial, especially if non-agent resources may have significant, autonomous dynamics.

4 Discussion and Conclusion

In this contribution, we proposed a first version of AMASON, an abstract core meta-model for MABS. Our goal was to suggest a meta-model that is generic enough to be able to capture a wide variety of agent-based simulation models. This idea poses a number of requirements, but also fixed starting points for our considerations: the concepts of agents shall be as simple as possible, but powerful enough to capture all forms of agents. Our basic agent concept is inspired by the *hysteretic* agents of Genesereth and Nilsson [11]; we wanted to give its internal state a structure in form of state variables in DEVS ([33]) as this is a clear and powerful generic concept. The distinction between body and mind is the result of a long discussion on how to capture the differences between passive objects or resources and active agents that we think must be made explicit providing ontological support. The structure of the environment in form of interconnected regions originates from a generalization of spatial representations observed in various MABS models. Interconnected regions are at first sight rather complex, yet this concept subsumes representations ranging from a single regions with a grid map to a network without any metric overhead. Additionally, it allows formulating environmental heterogeneity beyond heterogeneous populations.

AMASON is intentionally very simple, and in its current state covers only very basic structures and processes. Our focus was not on providing a highly elaborated meta-model with specific suggestions to all possibly occurring concepts, but one that fits best a broad variety of models in different domains, social science simulation and beyond. It contributes particularly to a clear conceptualization of embodiment and a generic spatial structure that can accommodate different types of spatial maps providing a heterogeneous environment for agent activities. Although AMASON is on high level of abstraction, we argue that it helps to clarify design decisions such as the granularity of actions that one has to take while designing a model.

AMASON is more basic than the AOSE meta-models mentioned above, which are more refined as they focus on providing languages for capturing interactions, coordination and organizations. On the other hand, our meta-model abstracts from particular data types or basic implementation-specific aspects that tool or programming language specific meta-models would include. A previous version of our meta-model was partially inspired by [16] with its focus on interaction between agent and its environment. AMASON is more basic and should also work for models in which the environmental dynamics model is not so essential.

In its current status, AMASON can already be used for "testing" platforms aligning their concepts to this meta-model, clarifying where the platforms deviated from the core agent ideas for providing easy to use tools. Moreover, with the definitions of AMASON, we argue that the teaching of MABS becomes easier, as the core concepts are more clear and independent from any particular tool.

Naturally, there are a number of aspects that we have not yet tackled, but are important for MABS. In addition to the context information, the main point is the missing explicit conceptualization of direct interactions between agents, agent relations, and organizational structures. This is clearly an extension of the `mind` that we will address in the future. The conceptualization of basic interaction is also not as concise and fully clear as it could be, especially with respect to interactions that influences multiple regions. This forms a problem that we have to solve as the next step.

References

1. Bandini, S., Manzoni, S., Vizzari, G.: Agent based modeling and simulation: An informatics perspective. J. Artificial Societies and Social Simulation 12(4) (2009)
2. Bernon, C., Cossentino, M., Gleizes, M.-P., Turci, P., Zambonelli, F.: A study of some multi-agent meta-models. In: Odell, J., Giorgini, P., Müller, J.P. (eds.) AOSE 2004. LNCS, vol. 3382, pp. 62–77. Springer, Heidelberg (2005)
3. Beydoun, G., Low, G., Henderson-Sellers, B., Mouratidis, H., Gomez-Sanz, J.J., Pavón, J., Gonzalez-Perez, C.: FAML: A generic metamodel for MAS development. IEEE Transactions on Software Engineering 35(6), 841–863 (2009)
4. Dinu, R., Stratulat, T., Ferber, J.: A formal approach to MASQ. In: van der Hoek, W., Kaminka, G.A., Lespérance, Y., Luck, M., Sen, S. (eds.) 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, CA, pp. 1443–1444 (May 2010)
5. D'Inverno, M., Luck, M.: Understanding Agent Systems, 2nd edn. Springer (2004)
6. Epstein, J.M., Axtell, R.: Growing Artificial Societies - Social Science from the Bottom Up. The MIT Press (1996)
7. Ferber, J., Gutknecht, O.: A meta-model for the analysis and design of organizations in multi-agent systems. In: ICMAS 1998: Proceedings of the 3rd International Conference on Multi Agent Systems, pp. 128–135. IEEE Computer Society, Washington, DC (1998)
8. Ferber, J., Müller, J.-P.: Influences and reactions: a model of situated multiagent systems. In: Proc. of the ICMAS 1996. AAAI Press (1996)
9. Fuentes-Fernandez, R., Galan, J.M., Hassan, S., Lopez-Paredes, A., Pavon, J.: Application of model driven techniques for agent-based simulation. In: Advances in Practical Applications of Agents and Multiagent Systems, PAAMS 2010, Salamanca, Spain (April 2010)
10. Garro, A., Russo, W.: easyABM: A domain-expert oriented methodology for agent-based modeling and simulation. Simulation Modelling Practice and Theory 18, 1453–1467 (2010)
11. Genesereth, M.L., Nilsson, N.J.: Logical Foundations of Artificial Intelligence. Morgan Kaufmann (1987)
12. Ghorbani, A., Bots, P., Dignum, V., Dijkema, G.: MAIA: a framework for developing agent-based social simulations. Journal of Artificial Societies and Social Simulation 16(2), 9 (2013)

13. Gilbert, N.: Agent-based Models. Sage Publications (2008)
14. Grimm, V., et al.: A standard protocol for describing individual-based and agent-based models. Ecological Modelling 198, 115–126 (2006)
15. Hahn, C., Madrigal-Mora, C., Fischer, K.: A platform independent meta model for multiagent systems. Autonomous Agents and Multi-Agent Systems 18(2), 239–266 (2009)
16. Helleboogh, A., Vizzari, G., Uhrmacher, A., Michel, F.: Modeling dynamic environments in multi-agent simulation. Autonomous Agents and Multi-Agent Systems 14(1), 87–116 (2007)
17. Hocaoglu, M.F., Firat, C., Sarjoughian, H.S.: Devs/rap: Agent-based simulation. In: Proc. of the 2002 AIS Conference (AI, Simulation and Planning in Highly Autonomous Systems), Lisboa (April 2002)
18. Klügl, F.: Towards a formal framework for multi-agent simulation models. Technical Report 412, Institute of Computer Science, University of Würzburg (2007)
19. Klügl, F., Fehler, M., Herrler, R.: About the role of the environment in multi-agent simulations. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) E4MAS 2004. LNCS (LNAI), vol. 3374, pp. 127–149. Springer, Heidelberg (2005)
20. Livet, P., Muller, J.-P., Phan, D., Sanders, L.: Ontology, a mediator for agent-based modeling in social science. Journal of Artificial Societies and Social Simulation 13(1), 3 (2010)
21. Macal, C.M., North, M.J.: Tutorial on agent-based modelling and simulation. Journal of Simulation 4, 151–162 (2010)
22. Müller, J.-P.: Towards a formal semantics of event-based multi-agent simulations. In: David, N., Sichman, J.S. (eds.) MAPS 2008. LNCS, vol. 5269, pp. 110–126. Springer, Heidelberg (2009)
23. North, M.J., Macal, C.M.: Managing BusinessComplexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation. Oxford University Press (2007)
24. Oechslein, C.: Vorgehensmodell mit integrierter Spezifikations- und Implementierungssprache zur Multiagentensimulation. PhD thesis, Computer Science Faculty, University of Würzburg, Germany (2003)
25. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the a&a meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems 17, 432–456 (2008)
26. Parker, M.T., North, M.J., Collier, N.T., Howe, T.R., Vos, J.R.: Agent-based meta-models. In: Sallach, D.L., Macal, C.M., North, M.J. (eds.) Proc. the Agent 2006 Conf. on Social Agents: Results and Prospects, pp. 69–79 (2006)
27. Railsback, S.F., Grimm, V.: Agent-based and Individual-based Modeling: A Practical Introduction. Princeton University Press (2011)
28. Siegfried, R., Lehmann, A., Khayari, R.E.A., Kiesling, T.: A reference model for agent-based modeling and simulation. In: Proceedings of SpringSim 2009, The 2009 Spring Simulation Multiconference, pp. 23:1–23:8 (2009)
29. Uhrmacher, A.M.: Object-oriented and agent-oriented simulation-implications for social science applications. In: Doran, J., Gilbert, N., Müller, U., Troitzsch, K.G. (eds.) Social Science Micro Simulation, pp. 432–447. Springer, Berlin (1996)
30. Urban, C.: PECS: A reference model for the simulation of multi-agent systems. In: Suleiman, R., Troitzsch, K.G., Gilbert, N. (eds.) Tools and Techniques for Social Science Simulation, pp. 83–114. Physica-Verlag (2000)
31. Weyns, D., Holvoet, T.: A formal model for situated multi-agent systems. Fundamenta Informaticae 63(2-3), 125–158 (2004)
32. Wooldridge, M.J.: Introduction to Multiagent Systems, 2nd edn. John Wiley & Sons (2009)
33. Zeigler, B.P.: Object Oriented Simulation With Hierarchical Modular Models: Intelligent Agents and Endomorphic Systems. Academic Press (1990)